Reinforcement Learning for Optimal Control of a District Cooling Energy Plant

Zhong Guo*,†, Austin R. Coffman*, and Prabir Barooah*,
*University of Florida.

***Abstract*—District cooling energy plants (DCEPs) consisting of chillers, cooling towers, and thermal energy storage (TES) systems consume a considerable amount of electricity. Optimizing the scheduling of the TES and chillers to take advantage of time-varying electricity price is a challenging optimal control problem. The classical method, model predictive control (MPC), requires solving a high dimensional mixed-integer nonlinear program (MINLP) because of the on/off actuation of the chillers and charge/discharge of TES, which are computationally challenging. RL is an attractive alternative: the real time control computation is a low-dimensional optimization problem that can be easily solved. However, the performance of an RL controller depends on many design choices.**

In this paper, we propose a Q-learning based reinforcement learning (RL) controller for this problem. Numerical simulation results show that the proposed RL controller is able to reduce energy cost over a rule-based baseline controller by approximately 8%, comparable to savings reported in the literature with MPC for similar DCEPs. We describe the design choices in the RL controller, including basis functions, reward function shaping, and learning algorithm parameters. Compared to existing work on RL for DCEPs, the proposed controller is designed for continuous state and actions spaces.

I. Introduction

In the U.S., 75% of the electricity is consumed by buildings, and a large portion of that is due to heating, ventilation, and air conditioning (HVAC) systems [1]. In cities and campuses, a large part of the HVAC's share of electricity is consumed in District Cooling Energy Plants (DCEPs). A DCEP produces and supplies chilled water to a group of buildings it serves, and the air handling units in those buildings use the chilled water to cool and dehumidify air before supplying it to building interiors. Figure 1 shows a schematic of such a plant, which consists of multiple chillers that produce chilled water, a cooling tower that rejects the heat to the environment, and a thermal energy storage system (TES) for storing chilled water to take the advantage of time-varying electricity price.

At present, DCEPs are typically operated by rule based control algorithms. But making the best use of the chillers and the TES to keep electricity cost at the minimum requires non-trivial decision making. Because of time coupling, especially due to the TES, the problem is best cast as an optimal control problem, of which the objective is the total electricity cost while meeting the thermal load from the buildings and equipment limitations are the constraints.

A growing body of work has proposed algorithms for optimal real-time control of DCEPs. This includes Model Predictive Control (MPC), such as [2]–[6]. MPC requires solving a high-dimensional nonlinear mixed integer program (MINLP) at every instant, since certain decision variables, such as chiller on/off and TES charge/discharge commands, are discrete and dynamics of the equipment in DCEPs are nonlinear. Solving a large MINLIP is computationally intractable. An MILP approximation is sometimes used, though solving a large MILP is also challenging.

An alternative to MPC is Reinforcement Learning (RL) which approximates an optimal policy from data collected from a physical system, or more frequently, its simulation. Despite a computationally burdensome learning phase, real-time control is simpler since control computation is an evaluation of a state-feedback policy. This advantage is particularly strong for problems involving discrete decision variables. RL is thus an attractive candidate for optimal control of DCEPs.

In this paper we propose an RL controller for a DCEP with multiple chillers, a cooling tower and a TES, with a goal of reducing energy cost while meeting the load from buildings. The proposed controller uses a batch RL algorithm similar to the "convex Q-learning" proposed in recent work [7] and the least squares policy iteration (LSPI) algorithm [8]. Closed loop simulations with a model calibrated using data from a Singapore campus shows a cost saving around 8% comparing to the baseline. This value is comparable to that of MPC controllers with MILP formulation reported in [2], [6]. Compared to MPC, the real time computation burden of the RL controller is trivial.

Designing RL controllers for practical applications with non-trivial dynamics is challenging. Firstly, its performance depends on myriad design choices: the choice of the state space, cost and reward, function approximation architecture and bases, learning algorithm and exploration method, etc. Secondly, training a RL controller is computationally intensive. Thirdly, if a particular set of design choices lead to a policy that does not perform well, there is no principled method to look for improvement. Although RL is being extensively studied in the control community, most works demonstrate their algorithms on plants with simple dynamics

† corresponding author, email: zhong.guo@ufl.edu.
The authors are with the Dept. of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32601, USA. The research reported here has been partially supported by the NSF through award 1934322 (CMMI) and 2122313 (ECCS).

with a small number of states and inputs [9], [10]. The model for a DCEP used in this paper, arguably still simple compared to available simulation models (e.g. [11]), is quite complex: it has 8 states, 5 control inputs, 4 disturbance inputs, and requires solving an optimization problem to compute the next state given the current state, control, and disturbance.

A number of papers have investigated the use of RL for control of HVAC systems; see [12]–[14] and references therein. The refs. [12], [13] are on control of air handling units (AHUs), which are considerably simpler than DCEPs. The refs. [15]–[17] are more relevant to the topic of this paper: control of a DCEP. In [15], [16] RL controllers to operate chillers are proposed using Watkins' Q-learning. The problem formulation is narrower; there is no TES, and the controller in [15] computes only one setpoint, the chilled water supply water temperature setpoint, while [16] computes two setpoints, operating frequencies of cooling tower fans and cooling water pumps. Another relevant work [17] also uses Q learning to train an controller that determines zone temperature setpoints and TES charge/discharge flow rates. However, trajectories of external inputs, e.g., outside air temperature and electricity price, are the same for all training days in [17]. Therefore, the RL controller needs to be retrained for every distinct external input trajectory.

The contributions of this work over the related literature on "RL for DCEPs" cited above are as follows. One, the proposed RL controller does not require discretizing the state space like the prior works [15], [16]. Two, it computes all five setpoints required to operate a DCEP (described in detail in Section II), compared to one or two as done in the prior works. Three, in contrast to works such as [17], we treat external inputs as time-varying disturbances and include them as RL states, making the proposed RL controller applicable to any time-varying disturbances, not just the trajectories considered during training.

The rest of the manuscript is organized as follows. Section II describes the DCEP, the control problem, and the simulation model. Section III describes the proposed controller, and Section IV provides its simulation-based evaluation. Section V concludes the paper.

II. System description and control problem

The DCEP under consideration is shown in Figure 1. The heat load from the buildings is absorbed by the chilled water supplied by the DCEP, and thus the return chilled water is warmer. The related variables in this process are denoted by superscript lw for "load water". The chiller loop (subscript ch) removes this heat and transmits it to the cooling water loop (subscript cw). The cooling water loop then sends the heat to the cooling tower, where the heat is rejected to the ambient. Connected to the chilled water loop is a TES tank that stores water (subscript tw). The total volume of the water in the TES tank is constant, but a thermocline separates two volumes: cold water that is supplied by the chiller (subscript twc for "tank water, cold") and warm water returned from the load (subscript tww for "tank water, warm").

A. DCEP dynamics

By discretizing time with a sampling period t_s, the control command for the DCEP at time step k is:

$$u_k = [n^{\text{ch}}_k,\ \dot{m}^{\text{lw}}_k,\ \dot{m}^{\text{tw}}_k,\ \dot{m}^{\text{cw}}_k, \dot{m}^{\text{oa}}_k]^T \in \mathsf{U}, \tag{1}$$

where n^{ch} is the number of active chillers, and $\dot{m}$ is mass flow rate of water, with superscripts describing the name of a water loop, except $\dot{m}^{\text{oa}}$, which is the mass flow rate of air passing through the cooling tower. Each of these variables can be independently chosen as setpoints since lower level PI-control loops maintain them. There are limits to these setpoints, which determine the admissible input set U:

$$\begin{aligned}\mathsf{U} \triangleq \{0,\dots,n^{\text{ch}}_{\max}\} \times [\dot{m}^{\text{lw}}_{\min}, \dot{m}^{\text{lw}}_{\max}] \times [\dot{m}^{\text{tw}}_{\min}, \dot{m}^{\text{tw}}_{\max}] \dots \\ \times [\dot{m}^{\text{cw}}_{\min}, \dot{m}^{\text{cw}}_{\max}] \times [\dot{m}^{\text{oa}}_{\min}, \dot{m}^{\text{oa}}_{\max}] \subset \mathbb{R}^5.\end{aligned} \tag{2}$$

The state of the DCEP is x^p (superscript p is for plant):

$$x^p_k \triangleq [T^{\text{lw,r}}_k, s^{\text{tww}}_k, s^{\text{twc}}_k, T^{\text{twc}}_k, T^{\text{tww}}_k, T^{\text{chw,s}}_k, T^{\text{cw,r}}_k, T^{\text{cw,s}}_k]^T, \tag{3}$$

where $s^{\text{tww}}, s^{\text{twc}}$ are the volumes of the warm water and cold water in the TES tank. The other state variables are temperatures at various locations; see Figure 1 (subscript "$,s$" for supply and "$,r$" for return) . The plant state x^p is affected by exogenous disturbances $w^p_k := [T^{\text{oawb}}_k, q^{L,\text{ref}}_k]^T \in \mathbb{R}^2$, where $q^{L,\text{ref}}_k$ is the cooling load needs to be removed from buildings, and T^{oawb}_k is the ambient wet bulb temperatures. There is a large literature on estimating and/or forecasting loads for buildings; see [18], [19] and references therein. We therefore assume $q^{L,\text{ref}}_k$ is known to the controller at k.

The control command and disturbances affect the state through a highly nonlinear dynamic model:

$$x^p_{k+1} = f(x^p_k, u_k, w^p_k), \tag{4}$$

that is described in the Appendix in [20].

B. Electrical demand and energy cost

The electricity cost at time k is:

$$c^{\text{E}}_k = t_s \rho_k P^{\text{tot}}_k, \tag{5}$$

where ρ_k ($\frac{\text{USD}}{\text{kWh}}$) is the electricity price and the total power consumption of the DCEP (P^{tot}) is the sum of that from chillers (P^{ch}), the cooling tower (P^{ct}), the chilled water pump ($P^{\text{cw,pump}}$), and the cooling water pump ($P^{\text{cw,pump}}$):

$$P^{\text{tot}}_k = P^{\text{ch}}_k + P^{\text{ct}}_k + P^{\text{chw,pump}}_k + P^{\text{cw,pump}}_k, \tag{6}$$

Details of the terms $P^{(\cdot)}$ can be found in [20].

C. The control problem

The control goal is to operate the DCEP in a way so that the cost is minimized while the cooling load $q^{L,\text{ref}}_k$ is met. The control problem for a scheduling horizon T is then:

$$\begin{aligned}\min_{\{u_k\}_{k=0}^{T-1}} \ & \sum_{k=0}^{T-1} c^{\text{E}}_k \\ \text{s.t.} \ & x^p_{k+1} = f(x^p_k, u_k, w^p_k),\ x^p_0 = x, \\ & q^L_k(x^p_k, u_k) = q^{L,\text{ref}}_k \text{ and } u_k \in \mathsf{U}(x^p_k, w_k).\end{aligned} \tag{7}$$

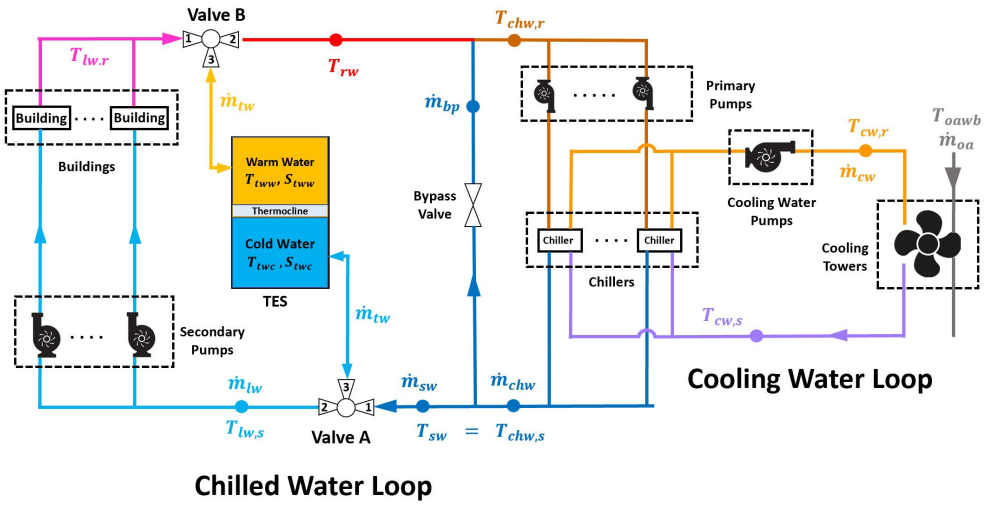

Fig. 1. Detailed description of District Cooling Energy Plant.

where $q_k^L(x_k^p, u_k)$ is the actual cooling load met by the DCEP, which is a complicated function of the states and inputs, and

$$\begin{aligned}\mathsf{U}(x_k^p, w_k) \triangleq \big\{u_k \in \mathsf{U}:\ & \dot{m}_k^{\text{lw}} \geq \dot{m}_{\min}^{\text{lw}},\ \dot{q}_k^{\text{ct,rej}} \geq \dot{q}_k^{\text{evap}},\\ & \dot{m}_k^{\text{lw}} + \dot{m}_k^{\text{tw}} \leq \dot{m}_k^{\text{chw}},\\ & T_{k+1}^{\text{cw,s},-} \leq T_{k+1}^{\text{cw,s}} \leq T_{k+1}^{\text{cw,s},+},\\ & S_{\min}^{\text{tw}} \leq S_k^{\text{twc}} + t_s \dot{m}_k^{\text{tw}} \leq S_{\max}^{\text{tw}}\big\},\end{aligned} \tag{8}$$

where $T_{k+1}^{\text{cw,s},-} = \max\left(25, T_{k+1}^{\text{oawb}} + 0.5\right)$ and $T_{k+1}^{\text{cw,s},+} = \min\left(T_k^{\text{cw,r}}, 40\right)$.

The solution of (7) provides the best achievable control performance over the horizon T. The optimization problem (7) is a high-dimensional MINLP due to n_k^{ch} being an integer (with the number of integer variables equal to the planning horizon T) and the nonlinear dynamics (4). Solving large MINLPs is intractable at present. In the following we propose a RL controller whose real-time computation burden is extremely small.

We omit the details of the dynamic model $x_{k+1} = f(x_k^p, u_k, w_k^p)$ here due to lack of space; the interested reader is referred to [20]. We note that the simulation model is highly nonlinear and complex: it requires solving an optimization problem to compute the function $f(\cdot,\cdot,\cdot)$.

III. RL Basics and Proposed Controller

A. RL basics

For the following construction, let x represent the state with state space X and u the input with input space $\mathsf{U}(x)$. Now consider the following infinite horizon discounted optimal control problem

$$\begin{aligned}J^*(\bar{x}) = \min_{\mathbf{U}}\ & \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k), \quad x_0 = \bar{x}, \\ \text{s.t.}\ & x_{k+1} = F(x_k, u_k),\ u_k \in \mathsf{U}(x_k),\end{aligned} \tag{9}$$

where $\mathbf{U} \triangleq \{u_0, \dots, \}$, $c : \mathsf{X} \times \mathsf{U} \to \mathbb{R}^{\geq 0}$ is the stage cost, $\gamma \in (0,1)$ is the discount factor, $F(\cdot,\cdot)$ defines the dynamics, and $J^* : \mathsf{X} \to \mathbb{R}^+$ is the optimal value function. The goal of the RL framework is to learn an approximate optimal policy $\phi : \mathsf{X} \to \mathsf{U}$ for the problem (9) without requiring explicit knowledge of the model $F(\cdot,\cdot)$. The learning process is based on the Q function. Given a policy ϕ for the problem (9), the Q function associated with this policy is defined as:

$$Q_\phi(x,u) = \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k), \quad x_0 = x, \quad u_0 = u, \tag{10}$$

where for $k \geq 0$ we have $x_{k+1} = F(x_k, u_k)$ and for $k \geq 1$ we have $u_k = \phi(x_k)$. A well known fact is that the optimal policy satisfies [21]:

$$\phi^*(x) = \arg\min_{u \in \mathsf{U}(x)} Q^*(x,u), \quad \text{for all} \quad x \in \mathsf{X}, \tag{11}$$

where $Q^* \triangleq Q_{\phi^*}$ is the Q function for the optimal policy. Further, for any policy ϕ the Q function satisfies the following fixed point relation:

$$Q_\phi(x,u) = c(x,u) + \gamma Q_\phi\big(x^+, \phi(x^+)\big), \tag{12}$$

for all $u \in \mathsf{U}(x)$ and $x \in \mathsf{X}$ and $x^+ = F(x,u)$. The above relation is termed here as the fixed-policy Bellman equation.

B. RL algorithm: data driven policy iteration

The learning algorithm has two parts: policy evaluation and policy improvement. First, in policy evaluation, a parametric approximation to the fixed policy Q function is learned by minimizing the residual error from (12). Second, in policy improvement, the learned approximation is used to define a new policy based on (11). For policy evaluation, suppose for a policy ϕ the Q function is approximated as:

$$Q_\phi^\theta(x,u) \approx \hat{q}_\theta(x,u), \tag{13}$$

where $\hat{q}_\theta(\cdot,\cdot)$ is the function approximator and $\theta \in \mathbb{R}^d$ is the parameter vector. To fit the approximator, suppose that the system is simulated for T_{sim} time step, then we can obtain the temporal difference error for the approximator:

$$d_k(\theta) = c(x_k, u_k) + \gamma \hat{q}_\theta(x_{k+1}, \phi(x_{k+1})) - \hat{q}_\theta(x_k, u_k), \tag{14}$$

for k = [0, T_{sim}-1]. We can then obtain θ^* by solving the following optimization problem:

$$\theta^* \stackrel{\Delta}{=} \arg\min_{\theta} \|D(\theta)\|_2 + \alpha\|\theta - \bar{\theta}\|_2, \tag{15}$$

where $D(\theta) \stackrel{\Delta}{=} [d_0(\theta), \ldots, d_{\mathsf{T}_{\text{sim}}-1}(\theta)]$. The term $\|\theta - \bar{\theta}\|_2$ is a regularizer and α is a gain. The values of $\bar{\theta}$ and α are specified in step 3) of Algorithm 1. The solution to (15) results in $Q_\phi^{\theta^*}$, which is an approximation to Q_ϕ. The quantity $Q_\phi^{\theta^*}$ can be used to obtain an improved policy, denoted ϕ^+, through:

$$\phi^+(x) = \arg\min_{u \in \mathsf{U}(x)} Q_\phi^{\theta^*}(x, u), \quad \text{for all} \quad x \in \mathsf{X}. \tag{16}$$

This process of policy evaluation (15) and policy improvement (16) can be repeated, which is described formally in Algorithm 1.

Algorithm 1: Data Driven Policy Iteration: Batch mode and off-policy

Result: An approximate optimal policy $\phi^{\mathsf{N}_{\text{pol}}}(x)$.
Input: T_{sim}, θ^0, N_{pol}, $\beta > 1$
for $j = 0, \ldots, \mathsf{N}_{pol} - 1$ **do**

1) Obtain input sequence $\{u_k^j\}_{k=0}^{\mathsf{T}_{\text{sim}}-1}$, initial state x_0^j, and state sequence $\{x_k^j\}_{k=1}^{\mathsf{T}_{\text{sim}}}$.

2) For $k = 1, \ldots, \mathsf{T}_{\text{sim}}$, obtain: $\phi^j(x_k) = \arg\min_{u \in \mathsf{U}(x_k^j)} \hat{q}_{\theta^j}(x_k^j, u)$.

3) Set $\bar{\theta} = \theta^j$ and $\alpha = \frac{j}{\beta}$ appearing in (15).

4) Use the samples $\{u_k^j\}_{k=0}^{\mathsf{T}_{\text{sim}}-1}$, $\{x_k^j\}_{k=0}^{\mathsf{T}_{\text{sim}}}$, and $\{\phi^j(x_k)\}_{k=1}^{\mathsf{T}_{\text{sim}}}$ to construct and solve (15) for θ^*.

5) Set $\theta^{j+1} = \theta^*$.

end

This algorithm is inspired by: (i) the Batch Convex-Q learning algorithm found in [7, Section III] and (ii) the least squares policy evaluation (LSPI) algorithm [8]. The approach here is simpler than the batch optimization problem that underlies the algorithm in [7, section III], which has an objective function that itself contains an optimization problem. In contrast to [8], our formulation ensures non-negativity of the Q-function.

C. Proposed RL controller for DCEP

We now specify the ingredients required to apply Algorithm 1 to obtain an RL controller for the DCEP from simulation data.

1) State space description: To define the state space for RL, we first denote w_k as the vector of exogenous variables:

$$w_k = [(w_k^p)^T, \rho_k, \bar{\rho}_k] \in \mathbb{R}^4. \tag{17}$$

where $\bar{\rho}_k = \frac{1}{\tau}\sum_{t=k-\tau}^{k} \rho_t$ is a backwards moving average of the electricity price with τ chosen to represent 4 hours. The expanded state for RL is:

$$x_k \stackrel{\Delta}{=} [x_k^p, w_k]^T \in \mathsf{X} \stackrel{\Delta}{=} \subset \mathbb{R}^{12}. \tag{18}$$

Note that all entries of x_k can be measured with commercially available sensors (e.g., T_{oawb}), or estimated from measurements (e.g., $q^{L,\text{ref}}$), or known via real-time communication (e.g., ρ_k and $\bar{\rho}_k$).

2) Design of stage cost: We wish to obtain a policy that tracks the load $q_k^{L,\text{ref}}$ whilst spending minimal amount of money, so we choose:

$$c(x_k, u_k) \stackrel{\Delta}{=} c_k^{\text{E}} + \kappa \left(q_k^L - q_k^{L,\text{ref}}\right)^2, \tag{19}$$

where $\kappa \gg 1$ to prefer load tracking over energy cost.

3) Approximation architecture: We choose the following linear-in-the-parameter approximation of the Q function:

$$Q_\phi^\theta(x, u) \approx \hat{q}_\theta(x, u) = \sum_{\ell=1}^{d} \psi_\ell(x, u)\theta_\ell, \tag{20}$$

where $\psi_\ell(x, u)$ are basis functions and $\theta \in \mathbb{R}^d$ is the parameter vector. We elect a quadratic basis and thus the approximation (20) can be equivalently expressed as:

$$Q_\phi^\theta(x, u) = [x, u] P_\theta [x, u]^T, \tag{21}$$

where P_θ is an appropriately chosen positive semidefinite matrix. The positive semidefiniteness is imposed since the Q-function is a discounted sum of non-negative terms (10).

4) Exploration strategy: We utilize a modified $\epsilon-$greedy exploration scheme. At time step k of iteration j, we obtain the input u_k^j from one of three methods: (i) the policy in step 2) of Algorithm 1, (ii) uniformly random feasible inputs, and (iii) the baseline controller described in Section IV-A. The choice to use either of the three controllers is determined by the probability mass function $\nu_{\text{exp}}^j \in \mathbb{R}^3$, which depends on the iteration index of the policy iteration loop:

$$\nu_{\text{exp}}^j = \begin{cases} [0, 0.1, 0.9] & \text{for } j \leq 5. \\ [0.5, 0.25, 0.25] & \text{for } j > 5. \end{cases} \tag{22}$$

The entries correspond to the probability of using the corresponding control strategy appeared in the (i)-(iii) order. The rational for this choice is that the BL controller provides "reasonable" state input examples for the RL algorithm in the early learning iterations so to steer the parameter values in the correct direction. After this early learning phase, weight is shifted towards the current working policy so to force the learning algorithm to update the parameter vector in response to its actions.

D. Real time implementation

Once the RL controller is trained, it computes the control command in real-time as:

$$u_k := \phi^*(x_k) = \arg\min_{u \in \mathsf{U}(x_k)} Q^{\hat{\theta}}(x_k, u), \tag{23}$$

where $\hat{\theta}$ is the parameter vector learned after N_{pol} policy improvements. Due to non-convexity of the set $\mathsf{U}(x_k)$ and integer nature of n_k^{ch}, the problem (23) is non-convex. To solve it, for each possible value of n_k^{ch}, we solve the corresponding continuous variable nonlinear program—a simple problem with a dimension of only four—using CasADi/IPOPT and then choose the minimum out of $(n_{\text{max}}^{\text{ch}} + 1)$ solutions.

TABLE I
SIMULATION PARAMETERS

Parameter	Unit	value	Parameter	Unit	value
β	N/A	100	γ	N/A	0.97
T_{sim}	N/A	432	N_{pol}	N/A	50
t_s	minutes	10	d	N/A	35
κ	N/A	500	θ_0	N/A	random
τ	hours	4	$\frac{S^{\text{twc}}_{\max}}{\rho_w}/\frac{S^{\text{twc}}_{\min}}{\rho_w}$	m^3	45/900
$n^{\text{ch}}_{\max}$	N/A	7	$\dot{m}^{\text{tw}}_{\max}/\dot{m}^{\text{tw}}_{\min}$	$\frac{\text{kg}}{\text{sec}}$	30/-30
$\dot{m}^{\text{lw}}_{\max}/\dot{m}^{\text{lw}}_{\min}$	$\frac{\text{kg}}{\text{sec}}$	350/20	$\dot{m}^{\text{cw}}_{\max}/\dot{m}^{\text{cw}}_{\min}$	$\frac{\text{kg}}{\text{sec}}$	300/20

*ρ_w is the density of water (kg/m^3).

IV. PERFORMANCE EVALUATION

A. Rule-based Baseline Controller

The RL controller's performance is compared to that of a rule-based baseline controller (BL) that determines $u_k = [n^{\text{ch}}_k,\ \dot{m}^{\text{lw}}_k,\ \dot{m}^{\text{tw}}_k,\ \dot{m}^{\text{cw}}_k, \dot{m}^{\text{oa}}_k]^T$ as follows. The quantity $\dot{m}^{\text{lw}}$ is determined based on the nominal temperatures of chilled water in the cooling coil (7 and 12 Celsius for $T^{\text{chw,s}}$ and $T^{\text{chw,r}}$, respectively). The TES water flow rate $\dot{m}^{\text{tw}}$is chosen based on a comparison between ρ and $\bar{\rho}$. If $\rho \leq \bar{\rho}$, the TES is charged at maximum flow rate until reaching its bound. Similarly, if $\rho \geq \bar{\rho}$, the TES is discharged at maximum flow rate until reaching its bound. The number of chillers n^{ch} is determined based on the required $\dot{m}^{\text{lw}}$ and $\dot{m}^{\text{tw}}$. The cooling water flow rate $\dot{m}^{\text{cw}}$is determined based on n^{ch} and $q^{L,\text{ref}}$. The air flow rate $\dot{m}^{\text{oa}}$ is determined based on the inlet conditions of the cooling tower to ensure proper operation.

B. Simulation setup

The parameters of the simulation model in Section II-A and electrical demand model in Section II-B are identified using data from the United World College of South East Asia Tampines Campus in Singapore [22], [23]. The cooling load and weather data are incorporated in the data set, see [20] for details. The real-time electricity price used in off-line learning and in real-time control is a scaled version of PJM's locational margin price during Sept. 6-12, 2021 [24]. Other relevant simulation parameters are located in Table I. The policy evaluation problem (15) is solved using CVX [25]. The optimization problems to update the policy and the DCEP dynamics are solved using CasADi/IPOPT [26], [27].

C. Numerical Results and Discussion

While both the RL controller and the baseline controller meet the cooling load requirement, the RL controller outperforms the baseline controller. The total electricity cost of the RL controller - \$2,051/week - compared to that of the baseline controller - \$2,214/week, reaches a saving of 8%. For comparison, savings by MPC controllers with mixed-integer formulation reported in the literature are 9.7% in [2] and 10.8% in [6].

Simulations are done for a week; the plots below show only two days to avoid clutter. *The results presented are "out-of-sample" results, meaning the external disturbance w_k used in the closed loop simulations are different from those used in training.*

The cost savings by the RL controller comes from its ability to use the TES to shift the peak electric demand

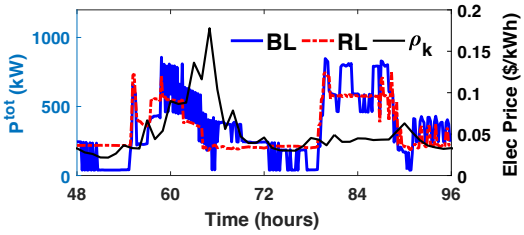

Fig. 2. Power consumption and real-time electricity price.

to periods of low price better than the baseline controller; see Figure 2. The cause for this difference is that the RL controller profits through the variation in the electricity price well, or at least better than the BL controller. This can be seen in Figure 3. The RL controller always discharges the TES during the peak electricity price while the baseline controller sometimes cannot do so because the volume of cold water is already at its minimum bound. The BL controller discharges the TES as soon as the electricity price rises, which may result in insufficient cold water stored in the TES when the electricity price reaches its maximum. While both the controllers use the same price information, the baseline controller cannot use that information as effectively as the RL controller.

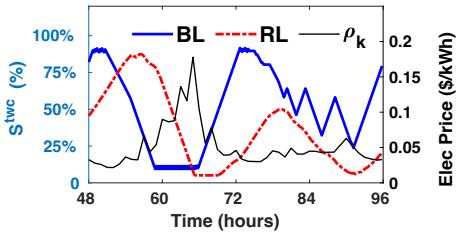

Fig. 3. TES cold water volume along with electricity price.

An alternate view of this behavior can be obtained by looking at the times when the chillers are turned on and off and the resultant supplied cooling, since using chillers cost much more than using the TES. We can see from Figure 4 that both BL and RL controllers shift the cooling they provide to the times when electricity is cheap. But the BL controller is not able to line up the supplied cooling with low price as well as the RL controller.

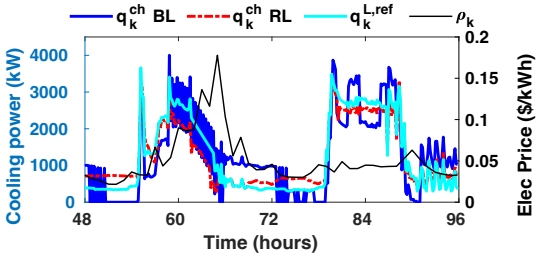

Fig. 4. Cooling load, cooling provided, and electricity price.

Another benefit of the RL controller is that it cycles the chillers less than the BL controller even the cost of switching between on-off status of chillers is not incorporated in the

cost function; see Figure 5. Fast cycling decreases the life expectancy of a chiller greatly.

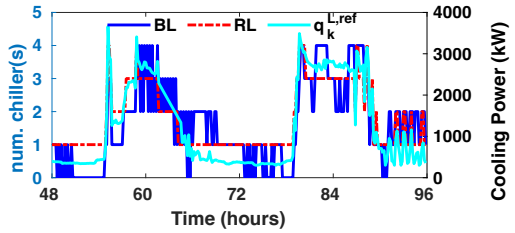

Fig. 5. Number of active chillers with respect to real-time electricity price for rule-based controller and RL controller

D. Lessons learned

Training of the RL controller is an iterative task that required trying many various configurations of the parameters appearing in Table I. In particular,

1) If the value of κ is too small, the controller will not learn to track the load $q_k^{L,\text{ref}}$. On the other hand, if κ is too large the controller will not save energy cost.
2) The choice of basis is critical. Redundant basis functions can lead to overfitting, which causes poor out-of sample performance of the policy. We avoid this effect by carefully selecting a reduced quadratic basis; see [20] for details. These choices were made based on physical and empirical intuition.
3) The condition number of (15) significantly affects the performance of Algorithm 1. The relative magnitudes of state and input values fundamentally determines the condition number. With scaling of the states/inputs, the condition number was reduced from 10^{20} to 10^3.

V. Conclusion

We proposed a RL controller for a district cooling energy plant that shows promise. The energy cost savings obtained by the proposed controller - 8% over baseline - is comparable to that obtained with MPC reported in other works, while requiring only a fraction of real-time computation.

In future work, we will test the robustness of the RL controller under different disturbance trajectories. We would also like to explore non-linear bases, such as neural networks, and examine convergence of the learning algorithm.

References

[1] "Commercial buildings energy consumption survey (CBECS): Overview of commercial buildings, 2012," Energy information administration, Department of Energy, U.S. Govt., Tech. Rep., December 2012.

[2] M. J. Risbeck, C. T. Maravelias, J. B. Rawlings, and R. D. Turney, "A mixed-integer linear programming model for real-time cost optimization of building heating, ventilation, and air conditioning equipment," *Energy and Buildings*, vol. 142, pp. 220 – 235, 2017.

[3] J. B. Rawlings, N. R. Patel, M. J. Risbeck, C. T. Maravelias, M. J. Wenzel, and R. D. Turney, "Economic MPC and real-time decision making with application to large-scale HVAC energy systems," *Computers & Chemical Engineering*, vol. 114, pp. 89–98, 2018.

[4] W. J. Cole, T. F. Edgar, and A. Novoselac, "Use of model predictive control to enhance the flexibility of thermal energy storage cooling systems," in *American Control Conference*, 2012, pp. 2788–2793.

[5] C. R. Touretzky and M. Baldea, "Integrating scheduling and control for economic MPC of buildings with energy storage," *Journal of Process Control*, vol. 24, no. 8, pp. 1292–1300, 2014.

[6] K. Deng, Y. Sun, S. Li, Y. Lu, J. Brouwer, P. G. Mehta, M. Zhou, and A. Chakraborty, "Model predictive control of central chiller plant with thermal energy storage via dynamic programming and mixed-integer linear programming," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 565–579, 2015.

[7] F. Lu, P. G. Mehta, S. P. Meyn, and G. Neu, "Convex Q-Learning," in *2021 American Control Conference (ACC)*. , 2021, pp. 4749–4756.

[8] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *The Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.

[9] G. Banjac and J. Lygeros, "A data-driven policy iteration scheme based on linear programming," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 816–821.

[10] B. Luo, D. Liu, H.-N. Wu, D. Wang, and F. L. Lewis, "Policy gradient adaptive dynamic programming for data-based optimal control," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3341–3354, 2017.

[11] C. Fan, K. Hinkelman, Y. Fu, W. Zuo, S. Huang, C. Shi, N. Mamaghani, C. Faulkner, and X. Zhou, "Open-source modelica models for the control performance simulation of chiller plants with water-side economizer," *Applied Energy*, vol. 299, p. 117337, 2021.

[12] C. Zhang, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Building HVAC scheduling using reinforcement learning via neural network based model approximation," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '19. Association for Computing Machinery, 2019, p. 287–296.

[13] N. S. Raman, A. M. Devraj, P. Barooah, and S. P. Meyn, "Reinforcement learning for control of building HVAC systems," in *American Control Conference*, July 2020, pp. 2326–2332.

[14] K. Mason and S. Grijalva, "A review of reinforcement learning for autonomous building energy management," *Computers & Electrical Engineering*, vol. 78, pp. 300–312, 2019.

[15] S. Qiu, Z. Li, Z. Li, and X. Zhang, "Model-free optimal chiller loading method based on q-learning," *Science and Technology for the Built Environment*, vol. 26, no. 8, pp. 1100–1116, 2020.

[16] S. Qiu, Z. Li, Z. Li, J. Li, S. Long, and X. Li, "Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation," *Energy and Buildings*, vol. 218, p. 110055, 2020.

[17] S. Liu and G. P. Henze, "Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory," *ASME Journal of Solar Energy Engineering*, vol. 129, p. 215–225, May 2007.

[18] J. E. Braun and N. Chaturvedia, "An inverse gray-box model for transient building load prediction," *HVAC&R Research*, vol. 8, pp. 73–99, 2002.

[19] Z. Guo, A. R. Coffman, J. Munk, P. Im, T. Kuruganti, and P. Barooah, "Aggregation and data driven identification of building thermal dynamic model and unmeasured disturbance," *Energy and Buildings*, vol. 231, p. 110500: 9 pages, January 2021.

[20] Z. Guo, A. R. Coffman, and P. Barooah, "Reinforcement learning for optimal control of a District Cooling Energy Plant," *arXiv preprint 2203.07500*, 2021.

[21] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[22] C. Miller, "united-world-colledge-open-data," https://github.com/buds-lab/united-world-college-open-data, 2014, [Online].

[23] C. Miller, Z. Nagy, and A. Schlueter, "A seed dataset for a public, temporal data repository for energy informatics research on commercial building performance," 06 2014.

[24] "PJM Interconnection Real-Time Hourly LMPs," https://dataminer2.pjm.com/feed/rt_hrl_lmps, 2021, [Online, accessed 2021-10-02].

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Feb. 2011.

[26] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, Mar 2019.

[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar 2006.